

EST 2016, 19° Concurso de Trabajos Estudiantiles

Simulación Colas M/M/1 en Arduino

Matías Haller¹, Silvia Cobialca²¹Estudiante, Universidad CAECE²Docente, Universidad CAECE

scobialca@yahoo.com.ar

Abstract. Por lo aprendido en la cursada de Modelos y Simulación, con asesoramiento de la profesora, se planteó en armar y programar un sistema de colas M/M/1, utilizando una placa Arduino conectado con sensores, servos y una pantalla led.

Keywords: Teoría de Colas, Arduino, Simulación

1 Introducción

Con este sistema se pretende simular y modelar la atención a clientes en un negocio o también en los procesos que se encolan en un servidor a la espera de ser atendidos, calculando el tiempo de atención y el tiempo promedio en el sistema de cada uno.

2 Teoría de Colas M/M/1

Las colas se presentan con frecuencia cuando se solicita un servicio por parte de una serie de clientes y tanto el servicio como los clientes son probabilísticos.

La teoría de colas es únicamente un modelo del comportamiento del tráfico que se ve todos los días, como lo puede ser un semáforo, la espera en un banco, la fila para conseguir el ticket para un concierto, así como el tráfico que representa el envío de paquetes en la red.

La teoría de colas presenta un panorama del comportamiento de la cola a través del tiempo y el entorno de la misma.

2.1 Concepto de colas

Un ejemplo de una cola es comprar un boleto para viajar, si existen pocas personas para ser atendidas, será una cola pequeña; sin embargo, si hay un gran número de personas esperando ser atendidas será una cola muy grande. Ahora bien, el número de cajas de atención dependerá de cuantas personas están atendiendo y el cliente será la persona que quiere comprar el boleto.

adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

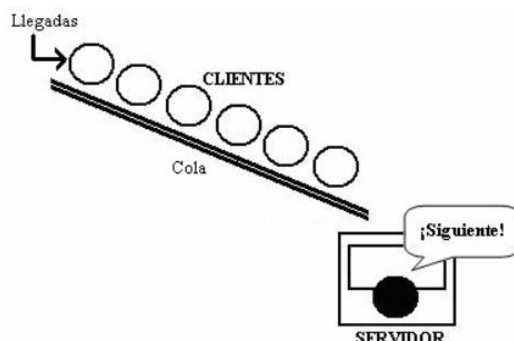


Fig. 1. Modelo de una sola cola con un único servidor

2.2 Características

Un sistema de colas se especifica por seis características principales.

1. El tipo de distribución de entradas o llegadas (tiempo entre llegadas)
2. El tipo de distribución de salidas o retiros (tiempo de servicio)
3. Los canales de servicio
4. La disciplina del servicio
5. El número máximo de clientes permitidos en el sistema
6. La fuente o población

2.3 Sistemas M/M/1

Considerando la notación de Kendall, para este sistema se tienen las siguientes características:

- (A) Se tiene un sistema de llegadas que se producen según un proceso de Poisson de razón λ , donde los tiempos entre llegadas estarán distribuidos exponencialmente $\text{Exp}(1/\lambda)$ o donde λ es el número medio de llegadas por unidad de tiempo
- (B) Los tiempos entre servicios son distribuidos de manera exponencial, $\text{Exp}(1/\mu)$ o donde μ es el número medio de paquetes o clientes que el servidor o caja de atención es capaz de atender por unidad de tiempo
- Se posee un único servidor en el sistema
- La capacidad del sistema es infinita, la cual se puede omitir
- La disciplina del sistema será FIFO, la cual se puede omitir
- Se tiene un estado de servicio igual a uno, es decir una sola cola, el cual se puede omitir también

Es decir, el sistema es el siguiente: $M/M/1/\infty /FIFO/1$, pero se abrevia como $M/M/1$. A continuación se irá analizando el sistema exclusivamente en su condición de no saturación, es decir como un estado estable, ya que si el sistema llega a saturarse el

número de paquetes en la cola crecerá indefinidamente, esto quiere decir que el sistema tendrá una tasa mayor de la que el servidor puede manejar.

3 Nuestro sistema

Se tomó el sistema de colas M/M/1, donde tendremos una cantidad finita de clientes con una sola caja de atención (servidor).

3.1 Funcionamiento

Representando las llegadas (Característica 1 o A).

Se coloca un carrito que contiene una cinta negra marcada con líneas blancas asociada a un motor servo. Cada una de estas líneas representa un cliente, la distancia entre las líneas se ha generado a partir de la distribución exponencial de media $1/\lambda$. Se designó una velocidad constante a los servos, de esta forma podemos simular el ingreso de clientes con una distribución de Poisson de razón λ .

La lectura de cada cliente se hace con un sensor RGB que detecta distintos tipos de colores. Al detectar la línea blanca se enciende un led Rojo y se escuchara un PITIDO con un parlante indicando el ingreso del cliente en el sistema. Este se encolara hasta ser atendido.

Representando la atención (Característica 2 o B).

Los tiempos entre servicios son distribuidos de manera exponencial, $\text{Exp}(1/\mu)$, también con una distribución de Poisson.

Para lograr esto se programó en el sistema un algoritmo que, a través de una distribución de Poisson de media μ ya asignada genera un tiempo aleatorio de atención.

Cada vez que se termina el tiempo de atención aleatorio genera uno nuevo con la distribución ya mencionada, e indica en la pantalla de LED el número de cliente siendo atendido.

3.2 Esperados

Los clientes ingresan al sistema (por medio del carrito marcado) simulando una distribución de Poisson con razón λ .

Una vez dentro del sistema, si la caja de atención está libre, el cliente ingresa a ser atendido mostrando en la pantalla LED el número de cliente, esto lo realizamos con un algoritmo que genera tiempos aleatorios de atención. En caso de que la caja de atención esté ocupada, los clientes se encolaran a la espera de ser atendidos.

Al terminar el carrito, es decir que no llegan más clientes, los clientes van a quedar encolados y serán atendidos a medida que se libere la caja de atención.

Si tenemos conectada una PC a la placa Arduino podemos ver en la consola el tiempo de espera de cada cliente y el tiempo de atención del mismo. Con estos valores se pueden obtener datos como:

- L = Número medio de clientes en el sistema,
- W = Tiempo medio que un cliente permanece en el sistema,
- W_q = Tiempo medio de espera en la cola,
- λ = Número medio de llegadas por unidad de tiempo,
- μ = Número medio de clientes que la caja de atención es capaz de atender por unidad de tiempo.

4 Arduino

4.1 Componentes

1. Placa Arduino MEGA 2680 II
2. Sensor RGB
3. Servo
4. Tornillería, ejes, engranajes
5. Pantalla LED 5x6
6. LED indicador de 5V color rojo.
7. Parlante o bocina.

4.2 Programación

La programación se hizo a través del IDE propio de Arduino, versión IDE 1.6.

Problemas.

A medida que se aprendía sobre la programación en Arduino encontramos dos problemas principales:

1. Poca memoria, haciendo que los cálculos de la distribución de Poisson se salgan de escala generando un stack overflow (desbordamiento de pila de memoria)
2. Al no tener un sistema operativo dentro del Arduino no se podían tratar los eventos de forma independiente (mediante threads). Normalmente en Arduino se programa con sleeps y loops para poder manejar eventos, eso hace que hay que esperar que termine el sleep o el loop para que se ejecute otro evento. Esto provoca que el sistema se pare por completo hasta que el evento termine.

En el sistema que estábamos tratando de simular este comportamiento no era deseable ya que tenemos clientes “arribando” al sistema constantemente y clientes que están siendo atendidos al mismo tiempo. También hay que tener en cuenta los cálculos que se realizan también en todo momento (distribución de Poisson, tiempo de llegada salida, etc.)

Soluciones.

1. Tuvimos que acotar la cuenta de la función de Poisson para que el Exp no sea tan alto, de esta forma pudimos evitar los problemas de memoria, dando como resultado un número muy cercano al que se necesitaba para simular el sistema.
2. Tuvimos que desarrollar un “clock”, tomamos los milisegundos desde que se encendía la placa para poder simular el paso del tiempo en nuestro sistema. Constantemente verificamos dichos milisegundos y verificamos los tiempos de atención, sin dejar de lado el “arribo” de nuevos clientes. Trabajando de esta forma pudimos emular un multi-threading.

5 Resultados

La siguiente tabla muestra los resultados obtenidos a partir de la simulación con el robot y su contraste con los valores teóricos:

Nro. De puestos de servicio: 1

Nro. De clientes en la cola al inicio: 3

Nro. De clientes en el Sistema: 3

Ciente	LAMBDA ARRIBOS	=	MU = SERVICIOS	Ciente	LAMBDA ARRIBOS	=	MU = SERVICIOS
26			00:00:05:64	13	00:00:03:00		00:02.8
25			00:00:04:77	12	00:00:04:30		00:10.7
24			00:04.5	11	00:00:00:92		00:10.4
23	0:00:02:96		00:04.8	10	00:00:03:06		00:07.6
22	0:00:01:14		00:07.1	8	00:00:04:06		00:07.9
21	00:00:01:69		00:08.7	7	00:00:03:11		00:07.1
20	00:00:01:86		00:09.3	6	00:00:06:64		00:06.1
19	00:00:02:39		00:05.0	5	00:00:03:67		00:05.9
18	00:00:01:73		00:09.8	4	00:00:01:49		00:06.7
17	00:00:03:43		00:08.0	3	00:00:06:88		00:07.3
16	00:00:02:12		00:07.2	2	00:00:01:51		00:08.2
15	00:00:02:77		00:10.8	001	00:00:06:52		00:05.0
14	00:00:08:14		00:06.1				

Basados en los datos de la tabla, se calcularon los parámetros y variables del sistema y a continuación se simuló a partir de los mismos un sistema similar, en lengua-

je R. A partir de la simulación se calcularon los mismos parámetros, se muestra a continuación una tabla comparativa de resultados:

Parámetro	Modelo con Robot	Simulación con R
M	1 (1 canal de atención)	1 (1 canal de atención)
λ	1 cliente cada 3 segundos	1 cliente cada 3 segundos
μ	1 cliente cada 7,4 segundos	1 cliente cada 7,4 segundos
$\rho = \lambda/\mu$	0,4054	0,4054
P (0)	0,594	0,594
L	0,6818	0,75
Lc	0,2764	0,34

6 Conclusión

Gracias al asesoramiento de la profesora y ayuda del laboratorio de robótica se pudo lograr una simulación de un sistema de colas M/M/1.

Este trabajo fue presentado a compañeros de clase, que tomando nota de los tiempos de arribo (PITIDO y encendido del LED rojo) y de atención (Muestra en pantalla de led número de cliente) pudieron sacar los datos para calcular:

L= Número medio de clientes en el sistema,

Lc= Número medio de clientes en la cola,

λ = Número medio de llegadas por unidad de tiempo,

μ = Número medio de clientes que la caja de atención es capaz de atender por unidad de tiempo.

Pudimos lograr/crear una representación de un sistema, diferente al sistema en sí. La utilización de la robótica permite extender la experimentación sin limitarnos a meramente lo experimental a través de un software sino que permite agregar más valor a dichos experimentos.

No solo pudimos entender el comportamiento del sistema o evaluar varias estrategias (dentro de límites impuestos por un criterio o conjunto de criterios) sino que además pudimos interactuar con un objeto físico, en este caso el robot, que nos permitió aprender nuevas maneras, más innovadoras de crear simulaciones sin desviarnos del objetivo principal.